Amendments to the Specification:

Please replace the first paragraph beginning on page 3 with the following amended paragraph:

In prior art graphics systems a texture memory contains a two dimensional representation of a texture to be mapped onto primitives. Cache provides temporary storage of portions of texture memory for improved access speed. Graphic primitives are stored in a ~~primitives~~ primitive storage portion of memory, and define the size and shape of graphic elements, such as triangles. Primitives are processed by triangle set up modules and traversed using edge walker modules and span walker modules. A texture mapping engine performs the operation of mapping textures stored in a texture memory onto primitives. Pixel processing is performed by 3D pipeline and performs operations on the pixels and writes the resulting rendered image via a render backend module to a frame buffer. The image in the frame buffer is sent to a display.

Please replace the paragraph beginning on page 6, line 29 with the following amended paragraph:

The graphic primitive can take the form of a triangle as depicted in FIGs. 4 and 5 of the drawings. As shown in FIG. 4, a primitive 400 has vertices at X0, Y0, X2, Y2 and [[X1Y1]]X1,Y1. The set up engine, such as set up engine 306 depicted in FIG. 3, determines the edge functions E0, E1 and E2 for the triangular primitive 400 wherein each edge function is associated with one particular edge of the graphic primitive. These edge functions are determined by the formulas depicted in FIG. 4 with the use of the three vertices, and the [[xy]]x,y variables for a pixel. Given the x,y coordinates of a particular pixel, if the three edge functions provide positive results, this means that the pixel is within the primitive 400. If any one of the edge functions provides a negative result it indicates that the pixel is outside of the primitive 400. The set up engine 306 also determines which of the edges is longest for the primitive 400 (this being the E0 edge function). The set up engine 306 also determines the starting vertices [[X0Y0]]X0,Y0 for the primitive.

Please replace the paragraph beginning on page 7, line 30 with the following amended paragraph:

Referring now to FIG. 6, the set up engine 600 provides slope information (such as, for example, for a color red), [[DR/DX]]dR/dx and [[DR/DY]]dR/dy. Registers 602 and 604 have inputs connected to the set up engine 600 for storing the slope information for the x and y position, respectively. Outputs of the registers 602 and 604 are connected to inputs of a first multiplexer 606. An adder 608 has a first input 610 connected to the output of the first multiplexer 606. A third register 612 stores a characteristic value (such as a red value) for a predetermined pixel. A second multiplexer 614 has a first input 616 connected to the output 618 of the third register 612, and has an output 620 connected to a second input 622 of the adder 608. A third multiplexer 624 has a first input 626 connected to the set up engine 600 and a second input 628 connected to an output 630 [[to]]of the adder 608. The output 632 of the third multiplexer 624 is connected to the input 634 of the register 612. These elements form the scan module described above.